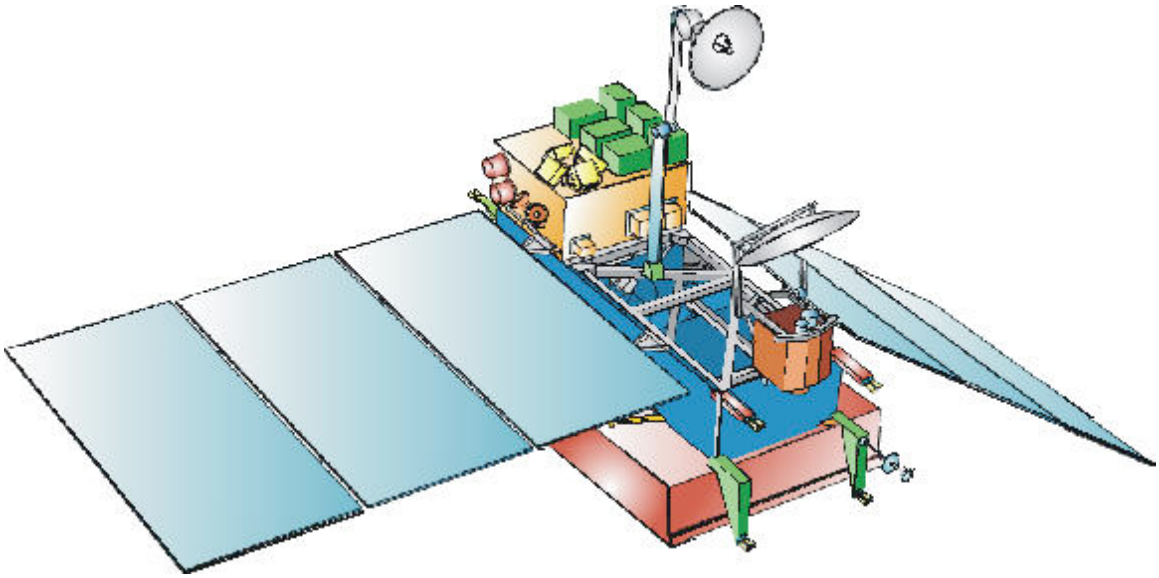
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GODDARD SPACE FLIGHT CENTER

White Paper

FSW IPSec Prospectus for GPM Spacecraft and Ground Systems



Global Precipitation Measurement (GPM)

Freemon Johnson NASA Goddard Space Flight Center
Flight Software Branch Code 582
Date: August 18, 2004

FOREWORD

I would like to thank the following participants for their cooperation and support across the various branches and industry in this effort. The following individuals I would like to recognize are as follows:

Team Members:

NASA:

1. Maureen Bartholomew Code 582 - ***GPM FSW Product Lead***
2. Jonathon Wilmot Code 582 – **Core FSW Executive & C&DH Lead**
3. Freemon Johnson Code 582 – **Communication Card Lead**
4. Carver Audain Code 582 – **FSW Software Systems Engineer**
5. Tim Rykowski Code 581 – **Operation and Ground Systems Manager**
6. Chris Silva Code 581 – **Ground Systems Engineer**
7. Charlie Wildermann Code 580 – **GPM Flight Data Systems Engineer**

Cisco (Cooperation Provided Under the Space Act Agreement):

1. Hugh Arif – **Space Initiatives Manager**
2. Natalie Timms – **Cisco IKE Working Group for IETF**
3. Scott Fanning – **Cisco IKE Working Group for IETF**

This team was not finished in its findings at the time of this document publishing because GPM engineering efforts were outsourced outside of NASA for completion.

The objective of this paper is to provide future missions who are not aware of IPSec's capability to secure TCP/IP based protocols for the purposes of command and telemetry of all mission space and ground assets.

Table of Contents

1. Executive Summary
2. Commanding Use Cases
    - 2.1 Nominal Commands
    - 2.2 Blind Commands
    - 2.3 T&C System Failure
    - 2.4 Communication Card Failure
    - 2.5. IPSec Operations During a Forward Link
3. IPsec Overview
4. Scope and Limitations
5. Mode and Phases of Operations
    - 5.1 Aggressive Mode
    - 5.2 Main Mode
    - 5.3 Quick Mode
6. Internet Key Exchange
    - 6.1 Version I.
    - 6.2 Version II.
    - 6.3 Version Summary
7. Security Algorithms
    - 7.1 Encryption Algorithms (Confidentiality)
    - 7.2 Hash Algorithms (Data Integrity and Host Authentication)
8. Conclusion
9. Appendix
    - 9.1. Performance Characterization
    - 9.2 References
    - 9.3 Vendor Specific Implementations
        - 9.3.1 OpenSwan/FreeSwan Timing Implementation
        - 9.3.2 Cisco's Timing Implementation
    - 9.4. Glossary
    - 9.5. Acronyms

1.0 Executive Summary

IPSec was considered to be used as a method for securing command telemetry to the spacecraft for the following reasons: 1.) Cost of implementation 2.) Well known and standardized by the IETF and the commercial sector. 3.) Approved by the NPR 2810.1 and future policy documentation both Goddard Policy and NASA Policy directives. and 4.) Has been proven by cryptologist to be the most secure method of transferring information over TCP/IP based networks. At the time of this study other methods considered prior were SSL/TLS, SSH, and DoD R/F equipment.

Unlike CCSDS protocols, TCP/IP is very well known. With all of the advantages associated with the utilization of commercial protocols you also inherit the vulnerabilities that comprise it as well. Research was needed in recognizing the differences and taking the necessary precautions to safeguard command and telemetry to and from the spacecraft. In lieu to this, ESTO provided funding through a proposal to produce an IP Security Handbook. This was the first document written in 2001 under the auspices of Code 588's OMNI Team, 584 and 290 that is now known to be Code 297 Enterprise Information and Security Brach. The initial contract requirements was describe in a GPG format of what would be needed to fly IP protocols safely for future mission that choose to utilize IP instead of CCSDS protocols. Several years later the Brunner Committee was formulated which comprised experts in the area of Space Asset Management and Protection and scientist and engineers abroad. The goal of this group was to make the center aware of the resources needed for the funding and expertise as we move forth to the next generation of communication with TCP/IP protocols for spacecraft communications. As of today it is uncertain what is written in the NPR2810.x that actually addresses onboard spacecraft security with respect to TCP/IP protocols.

The original NPG2810.1 focused on ground systems and not spacecraft subsystems. Even though the naming convention has changed to the NPR prefix, the content of the document thus far has not. The document has not been publicly made available presently. The NPG2810.1 did not go in any level of granularity surrounding the features that must be in place for the spacecraft to not be compromised itself. The former standard stated briefly that sensitive communications to the spacecraft should be encrypted in simplistic terms. This is not sufficient by today's policies to secure spacecraft from threats especially since in the same documentation it states you can accept the level of risk and "waive" your rights to secure your data if approved by the project management of the mission through headquarters administration. Because of past events and increased scrutiny on the protection of NASA Space Assets such as a spacecraft and all instruments on board, past directives are simply unacceptable and insufficient.

This document shows how Flight Software plans to harvest the security features of IPSec for the purposes of safe commanding and telemetry for GPM's spacecraft and ground system assets for TCP/IP based communications.

2.0 Commanding Use Cases

The premise of this section is to describe the cases that IPsec will be utilized in flight and ground operations. It is important to see what impact this security protocol will have on routine to critical communications from the MOC/POC to the spacecraft.

2.1 Nominal Commands

This section presents "use cases" describing operations concepts for how IPSec will be utilized to support GPM commanding operations.

Use Case: GPM Nominal commanding operations

Preconditions:

Use case is executed during all TDRSS SSA scheduled services

Use case is executed during all TDRSS MA-F scheduled services, under nominal MA-R operating conditions.

Use Case steps:

1. MOC sends message to Core spacecraft to initiate a secure tunnel, by delivering a ISAKMP security association proposal and a session key value.
2. Core spacecraft delivers message to MOC authenticating the MOC.
3. MOC sends message to Core spacecraft authenticating the Core spacecraft. At this point, a secure tunnel is established with the core spacecraft. MOC stores in continuity database all required security associations for future blind commanding if necessary.
4. MOC sends directives to Core spacecraft to kill/delete any previously established secure tunnels/security associations that may still be in existence.
5. MOC sends command stream to Core spacecraft, using the Encapsulating Security Payload tunnel mode formats as described in the GPM Core spacecraft Space-to-Ground ICD, Section 3.4.
6. ESP packets are encrypted using NIST-compliant algorithm TBD.
7. COP-1 protocol is in effect for command stream

2.2 Blind Commands

Use Case: GPM Blind commanding operations

Preconditions:

Use case is executed during TDRSS SSA or TDRSS MA-F events, where no return link data is present. Absence of return link signal attributed to a spacecraft anomaly or added pass, not previously reflected in the spacecraft's communications schedule.

Use Case Steps:

1. MOC analyzes security associations stored in Continuity database to identify command tunnels that have been previously established.

2. If a previously established security association exists, MOC sends command stream to the Core spacecraft, using the ESP tunnel mode formats described in the GPM space-ground ICD
3. ESP packets are encrypted using NIST-compliant algorithm TBD.
4. COP-1 protocol is NOT in effect for command stream.
5. If no previously established security association exists, MOC sends unencrypted command stream to the Core spacecraft. (Note: This condition would be satisfied upon time expiration of the last established security association).

2.3 T&C System Failure

Use Case: GPM failover commanding operations – prime T&C system failure

Preconditions:

1. GPM T&C system failure detected by appropriate software in MOC.
2. Sufficient time remains in event to deliver planned command stream to the Core spacecraft.

Use Case Steps:

1. MOC initializes backup T&C system for operations. (This activity should be accomplished within seconds).

2. MOC sends message to Core spacecraft to initiate a secure tunnel, by delivering an ISAKMP security association proposal and a session key value.

3. Core spacecraft delivers message to MOC authenticating the MOC.

4. MOC sends message to Core spacecraft authenticating the Core spacecraft. At this point a secure tunnel is established with the core spacecraft.

5. MOC sends directives to Core spacecraft to kill/delete any previously established secure tunnels that may still be in existence.

6. MOC executes procedure to resynchronize/reset sequence counters associated with COP-1 frames to the NES number anticipated by the Core spacecraft.

7. MOC sends command stream to Core spacecraft, using the Encapsulating Security Payload tunnel mode formats as described in the GPM Core spacecraft Space-to-Ground ICD, Section 3.4.

8. ESP packets are encrypted using NIST-compliant algorithm TBD.

9. COP-1 protocol is in effect for command stream.

10. Command transmission begins with the first command in the original stream/load (commands that were previously sent in the session prior to T&C system failure are re-delivered).

2.4 Communication Card Failure

Use Case: GPM failover commanding operations – communications card failure

Preconditions:

GPM MOC detects downlink comm. card failure through absence of downlink telemetry during scheduled event, and other troubleshooting activities/commanding activities to recover the spacecraft telemetry have been attempted, but have not succeeded.

Use Case Steps:

1. MOC sends a special hardware command to designate the alternate communications card as "prime".
2. MOC sends message to Core spacecraft to initiate a secure tunnel, by delivering a ISAKMP security association proposal and a session key value.
3. Core spacecraft delivers message to MOC authenticating the MOC.
4. MOC sends message to Core spacecraft authenticating the Core spacecraft. At this point a secure tunnel is established with the core spacecraft.
5. MOC sends directives to Core spacecraft to kill/delete any previously established secure tunnels that may still be in existence.
6. MOC executes procedure to resynchronize/reset sequence counters associated with COP-1 frames to the next expected sequence number anticipated by the Core spacecraft.
7. MOC sends command stream to Core spacecraft, using the Encapsulating Security Payload tunnel mode formats as described in the GPM Core spacecraft Space-to-Ground ICD, Section 3.4.
8. ESP packets are encrypted using NIST-compliant algorithm TBD.
9. COP-1 protocol is in effect for command stream.
10. Command transmission begins with the first command in the original stream/load (commands that were previously sent in the session prior to the anomaly are re-delivered).

2.5 IPSec Operations During a Forward Link Session

When an SA is created the sequence number is initialized to zero and prior to IPsec output processing, the value is incremented.

New SAs must be created prior to the sequence number wrapping around back to zero. The sequence number is 32 bits long so wrap-around is 2^{32} packets. The receive window can be any size greater than 32, however 64 is recommended. The window size should be a multiple of the size of a word on the computer on which IPsec is being implemented. So if a word is 4 bytes, 8 bytes should be the size of the receive window for example.

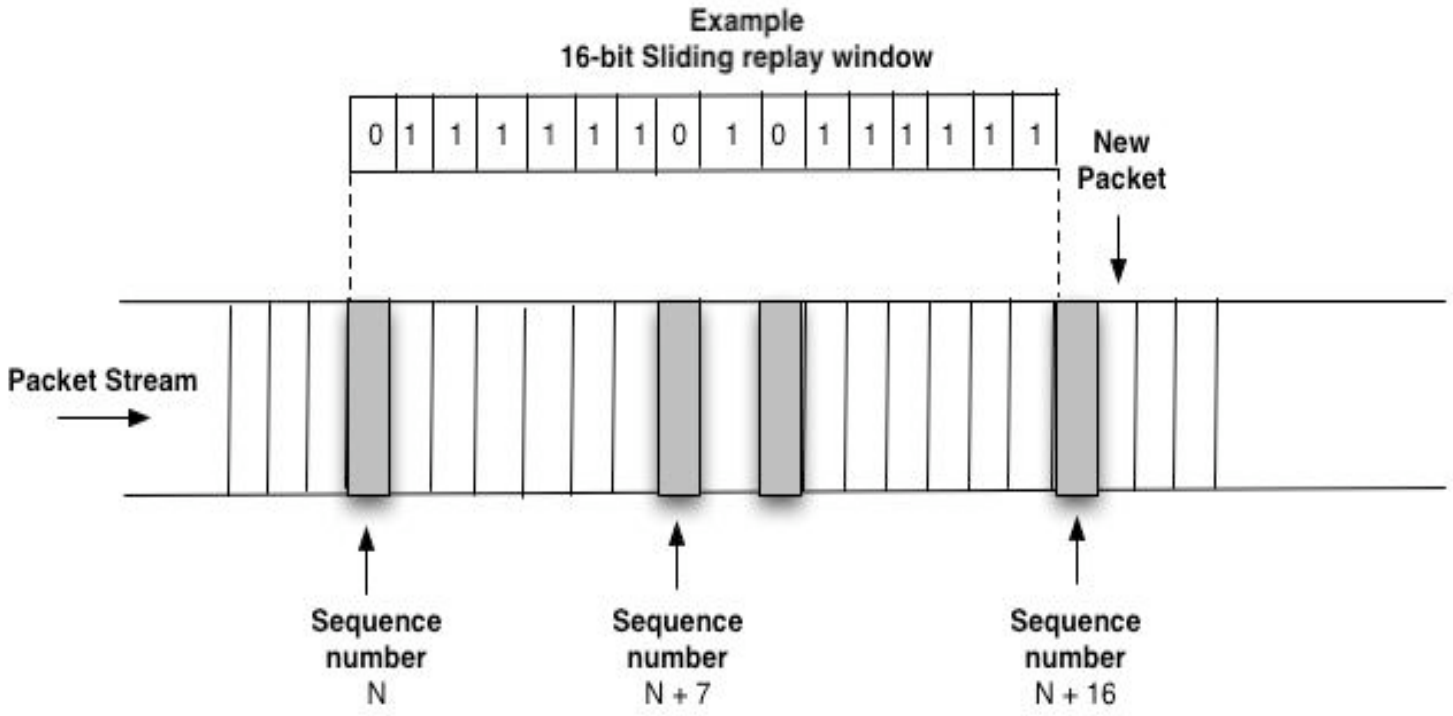
The received packet must be new and must fall either inside the window or to the right of the window, otherwise the packets are dropped. If a packet that is received is to the right of the window, it may also be dropped if it fails the authenticity test. If it passes the test the window advances to the right, to encompass the next packet. If a packet is received after a valid packet with a sequence number greater than the size of the window, the packet will be dropped.

The window must not be advanced until the packet that would cause its advancement has been authenticated. Doing otherwise would allow an attacker to generate bogus packets with large sequence numbers that would move outside the range of valid sequence numbers and cause the dropping of valid packets.

IPSec does not replace or augment forward sequence verification protocols such as COP-1. As stated prior IPSec resides at the network layer while COP-1 is at Layer 2 predominantly the Link Layer. This is also infers

that IPSec does not replace COP-1 nor does it inhibit COP-1 from functioning. COP-1 also does not inhibit IPSec from functioning properly as well. No re-engineering is needed or modification of the protocol in order to have COP-1 and IPSec coexist. Therefore, COP-1 is still needed for most mission to provide the sequencing and verification support at the link layer especially since this is tied into CLCW packets to verify packet arrival traditionally. To reemphasize as long as COP-1 or SLE (CCSDS Space Link Extension) does not merge the link layer and the network layer together, IPSec will function independently of the physical and framing layers below it.

This diagram is the example of a 16-bit sliding replay window. This size is invalid but I am displaying it for demonstration purposes only!



3.0 IPsec Overview

IPSec would be utilized in the use cases prior in section 2.0. IPSec will be used to provide an security association (SA) from the ground to the spacecraft. A security association consist of a configuration that contains the protocols used to transmit the packets, the transforms, the encryption keys, and the duration in which the keys are valid. SAs are unidirectional. Therefore, two host A and B are communicating securely using ESP (Encapsulating Security Protocol), one SA-out is used for egress packets and one SA-in for ingress packets to the interface for bidirectional communication assuming if both parties wanted to receive traffic and send it via IPSec. SAs are stored in a Security Association Database (SADB). They SADB is grouped by the SPD (Security Policy Database). The Security Parameter Index (SPI) issued an used as an index into the SPD and SADB to define what security association is being referenced once communication to the target has been established.

IP Security is a protocol developed with several implementations to provide authentication of the host, confidentiality of the user data, and integrity of the data. Pertaining to the layer 3 of the OSI model which is the network layer, the network protocol that IPsec pertains to is of course IP (Internet Protocol) hence the name IP Security. IPsec is independent of the transport layer 4 or application layers 7. It is also independent of the lower layers 1-2, which defines the transmission media, bit framing, etc. However, IPsec can be implemented between layers 2-3 using the BITS (Bump in the Stack) implementation. This is very vendor specific. As stated prior

IPsec is at layer 3 with IKE being at layer 4. I should also mention it is independent of the OASIS layers 8-15 which encompasses the application levels of data object framing and content management e.g. XML, etc.

IPsec guarantees the three attributes mentioned prior for a host-to-host, host-to-gateway, or gateway-to-gateway. This defines the two types of ways IPsec is configured via Tunnel Mode or Transport Mode. Transport Mode guarantees host authentication, data integrity, and anti-replay. Tunnel Mode guarantees all that was mentioned prior with the addition of confidentiality, which implicates encryption.

The protocol features that will permit this behavior is the AH (Authentication Header) and ESP (Encapsulating Security Protocol). Lets start first with the most robust of the two protocols that make Transport Mode and Tunnel Mode feasible which is ESP. ESP provides confidentiality, data integrity, and data source authentication of IP packets. It also provides protection against replay attacks. This is performed by inserting a new ESP header after the IP header (and any IP options), before the data being protected, appending an ESP trailer. ESP is defined by RFC 2406. Within a Security Association you can configure a cipher for confidentiality and an authenticator for authentication. The ESP header travels in the clear which contains the SPI used to associate which SA is to be used. The trailer contains the padding if any, length of the pad, the next protocol after the data if any.

The order of execution is as follows for ESP:

1. First verify the sequence number.
2. Then verify the integrity of the packet.
3. Then decrypt the data.

Since the sequence number and authentication is performed last, that information must be transmitted in the clear. Since encryption has to be performed in fixed block sizes, CBC (Cipher Block Chaining) mode is always used as opposed to stream ciphering i.e. RC4. CBC dictates that the amount of data to encrypt be a multiple block size of the cipher being used to encrypt the data hence why authentication keys are double that of the encryption key lengths stated prior. CBC also uses an initialization vector IV to “jump-start” the encryption process. Ninety-Six bits including the padding ensures alignment with IPv6.

The Authentication Header protocol provides data integrity, data source authentication, and protection against reply attacks. It does not provide confidentiality. Therefore, there is no encryption. AH is just a header and not a header and a trailer. Also all of AH data is transmitted in the clear. The RFC that defines AH is RFC2402. For Cipher suites, for HMAC-MD5-96 is defined for AH in RFC 2403 and for HMAC-SHA-96 in RFC 2404.

Replay is provided for AH and ESP protocols. Both a sequence number and a sliding receive window is used. Note that packets can be received out of order and still be processed properly.

More about Tunnel Mode and Transport Mode is as follows: Transport Mode AH and ESP intercept the packets flowing from the transport layer into the network layer and provide the configured security. AH or ESP protect the transport layer or encrypt the transport layer’s payload. Tunnel Mode is used the when the ultimate destination of the packet is different from the security termination point i.e you terminating the IPSec SA at a gateway instead of a host. Two IP headers are assumed in tunnel mode because one header is for the gateway’s IP address and the other IP header is for the host behind the gateway.

Packet formatting:

Transport Mode (the data integrity should be calculated over as much of packet as possible which is why AH is first instead of ESP. Also ESP is optional in this mode):

IP -> AH -> ESP - > TCP/UDP Header -> Data

Tunnel Mode (encrypts the internal IP address and data, not the gateway address. ESP is NOT optional in this mode. ESP encrypts the 2nd IP header and the TCP/UDP packet in its entirety):

IP Header 1-> ESP -> IP Header 2 - > TCP/UDP Header - > Data

The packet layout is defined as follows:

Table B-1: IPSec Packet Construct

PACKET CONSTRUCT	Authentication Header	Encapsulating Security Payload
Transfer Mode	*IP Header 1* *AH Header* *DATA*	IP Header 1 *ESP Header* ***DATA*** ***ESP Trailer*** ESP Authenticator
Tunnel Mode	*IP Header 2* *AH* *IP Header1* *DATA*	IP Header 2 *ESP Header* ***IP Header 1*** ***DATA*** ***ESP Trailer*** ESP Authenticator

- Italic denotes authenticated portions of a packet.
- Bold denotes encrypted portions of a packet.
- IP Header 1 is the original IP header information in a packet.
- IP Header 2 is the IP header added by IPSec, including the source and destination addresses listed as the endpoints of the IPSec tunnel.
- AH Header is header information added by the AH protocol.

4. Scope and Limitations

At the time of the writing of this document, the following was unfinished: IKEv.2 and NPR 2810.1 GPM FSW decided to baseline our conclusion on the IKEv.1 since it is mature and has been implemented commercially in a myriad of products. The NPG 2810.1 was also our baseline in accordance with the IP in Space Security Handbook and the NITR-2810-2.

In knowing this, each mission project team should familiarize themselves with the latest security policy that has been approved and distributed i.e. NPR 2810.1 as well as the RFC or RFC's that are associated with the IKEv.2 once the IETF has completed their working draft.

From NASA's standpoint there will be no backwards compatibility to my knowledge to the NPG documentation. Only the new rules and regulations will now set the precedence. As for the IKEv.2 since it is vendor specific with regards to its implementation, a vendor has the choice to or not to support legacy ISKMP protocols such as IKEv.1 The reader should be aware also that there is no mention of backwards compatibility or interoperability with IKEv.1 in the RFCs to come that will define IKEv.2. It is also up to the vendor to decide if their network premise equipment will support both protocols. It is IETF's decision that the protocol was difficult to comprehend because of the various modes and phases of operations and in lieu to this it was difficult to implement because of this. There were also security vulnerabilities associated with IKEv.1. IKEv.1 was proven to be weak for example in DoS attacks. How IKEv.2 addresses these issues is not disclosed in this document. Please refer to the RFC draft in the Reference section.

Our baseline for flight software is based on Openswan, which is the successor to Freeswan implementation for IPsec. This software implementation was proven to be in accord with RFCs mentioned for IKEv.1 and IPsec protocols. The source code is also open source under the GPL licensing as of this date. This is of course one vendor's implementation but in the Open Source community this is the most popular, robust and supported implementation to date. Freeswan in the past is also compatible with Cisco Pix product family as well as be RFC compliant. FSW would modify this code without breaking the RFC statutes as needed to integrate into our flight software environment.

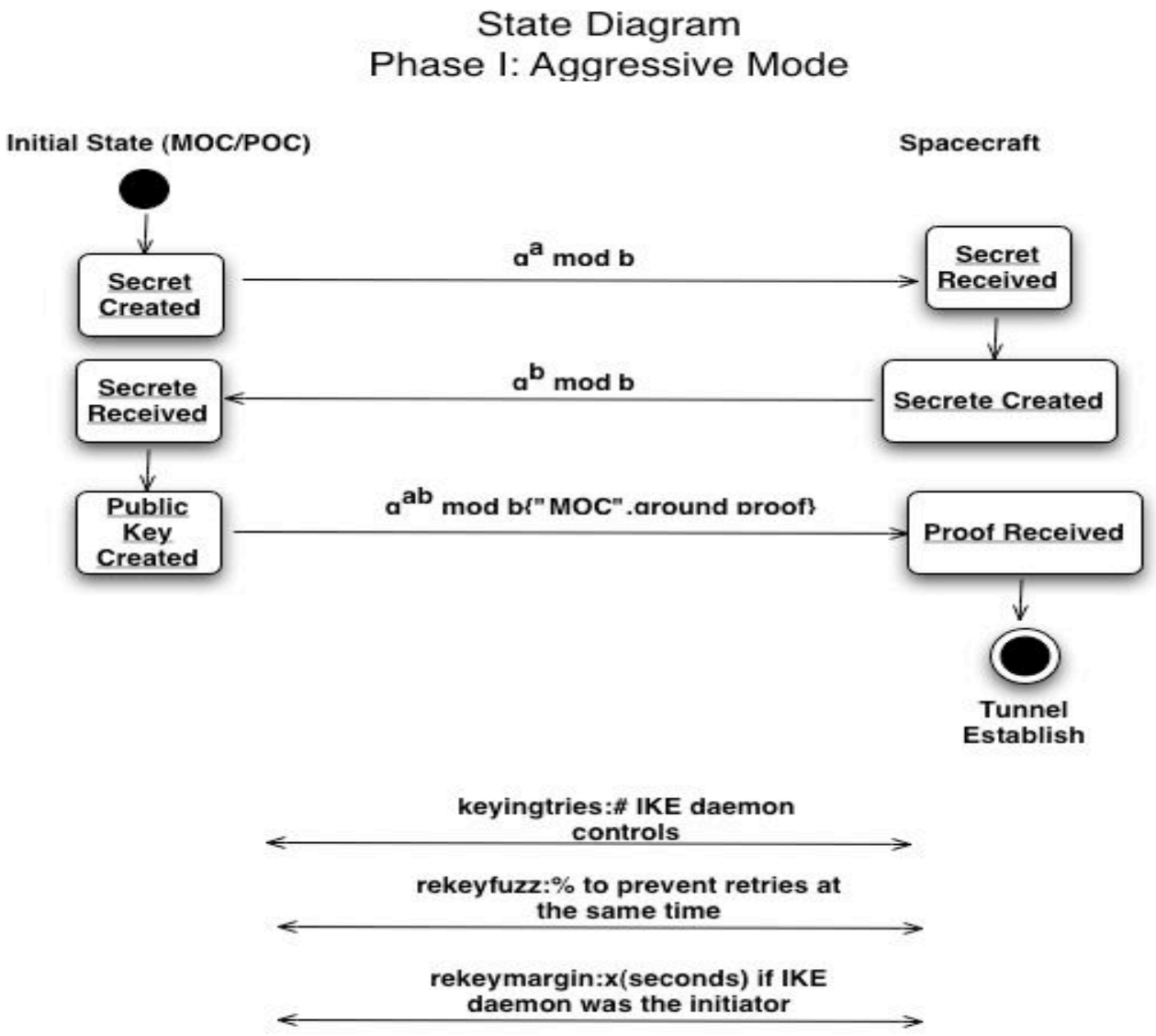
5. Modes and Phases of Operations

Even though FSW and ground systems chose Main Mode for GPM, it is necessary to understand the differences and why this decision came to be.

5.1 Aggressive Mode

This mode accomplishes mutual authentication and session key establishment in 3 messages. The fist two messages include a Diffie-Hellman exchange to establish a key session and in the second and third messages each side proves they know both the Diffie-Hellman value and their secret. Some key points are as follows:

- It was recommended by Cisco that you should use “strong” pre-shared keys with Aggressive Mode as a precaution at best.
- Aggressive Mode is "faster" because of the number of message needed to complete Phase I and II are less. Until we run test in the lab, the metric for "faster" is conceptual.
- Aggressive Mode's Phase II does not have to be originated from the source. Protocols such as RIP and OSPF can be used to negotiate Phase II. Main Mode does not support this.
- AH protocol only uses pre-defined Group # for establishing an SA e.g. Group 1, 2, etc. This is not to be confused with Diffie-Hellman Group numbers. Each Group has parameters that are predefined for an SA configuration.
- The destination can reject an establishment and not even notify the source what Groups it actually supported.
- There is no way to negotiate the group number for the Diffie-Hellman exchange.
- As for the cryptographic algorithms a side from Message #1 everything else can be negotiated.



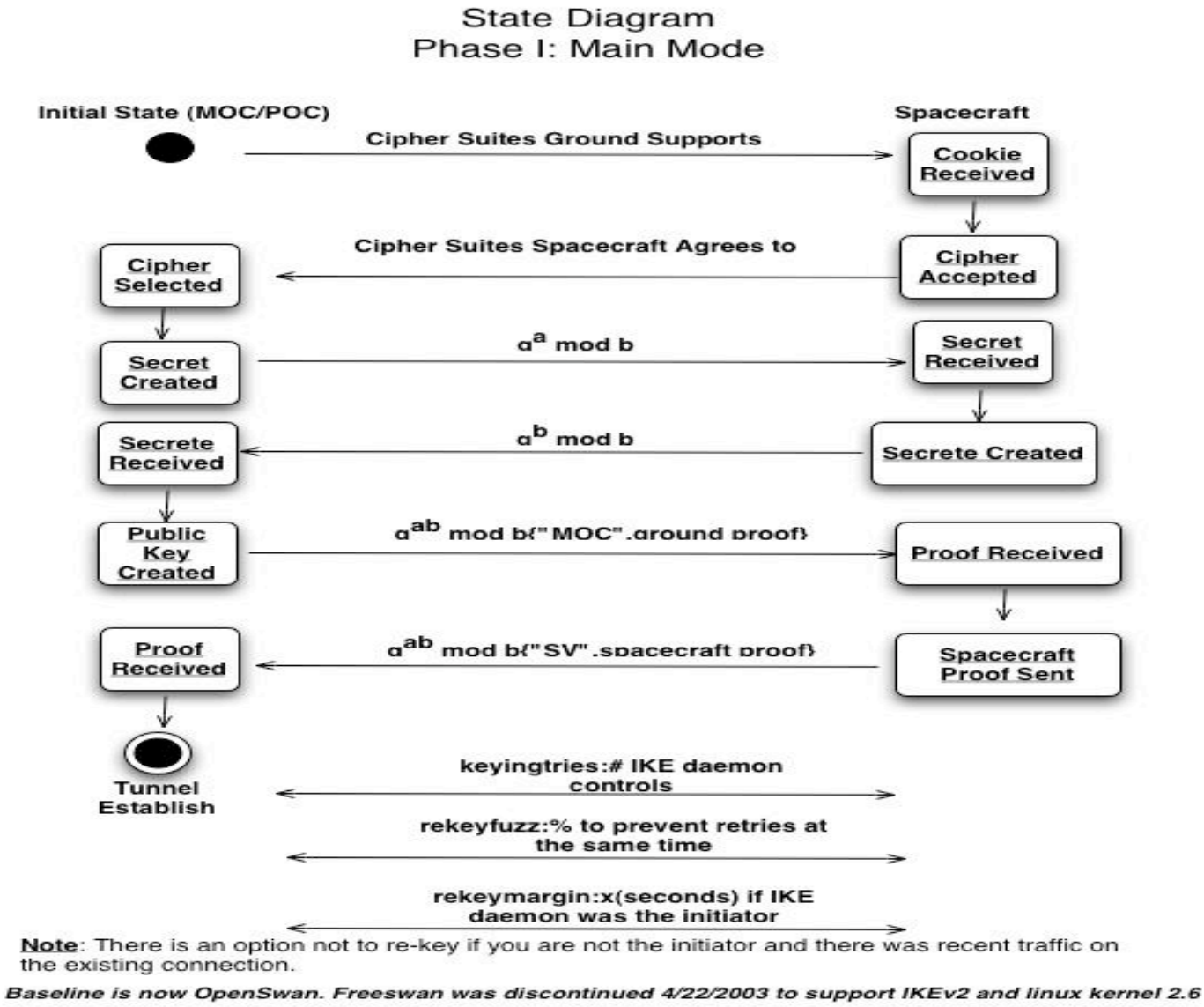
Note: There is an option not to re-key if you are not the initiator and there was recent traffic on the existing connection.

Baseline is now OpenSwan. Freeswan was discontinued 4/22/2003 to support IKEv2 and linux kernel 2.6

5.2 Main Mode

Uses 6 messages and has additional capabilities, such as the ability to hide endpoint identifiers from eavesdroppers and additional flexibility in negotiating cryptographic algorithms. In the first two messages for example Alice sends a cookie and requested cryptographic algorithms, and Bob responds with his cookie and the cryptographic algorithms he will agree to support. Messages 3 and 4 are a Diffie-Hellman exchange. Messages 5 and 6 are encrypted with the Diffie-Hellman value agreed upon in messages 3 and 4. Also in message 5 and 6 each side reveals its identity and proves it know the relevant secret e.g. private signature keys, or pre-shared secret. In short Main Mode:

- Main Mode offers admission control.
- Is acclaimed to be most secure for it provides identity protection. Aggressive Mode does not.
- Is more flexible and does not use pre-defined Group # to an establish an SA. You can customize the cipher suite and other parameters so as long as the receiver supports them.
- Diffie-Hellman is used in EVERY exchange during a SA instantiation regardless of the static keys or pre-defined configurations. Why? Because there are a myriad of options and the protocol has to be flexible to enough to satisfy all of the requirements.

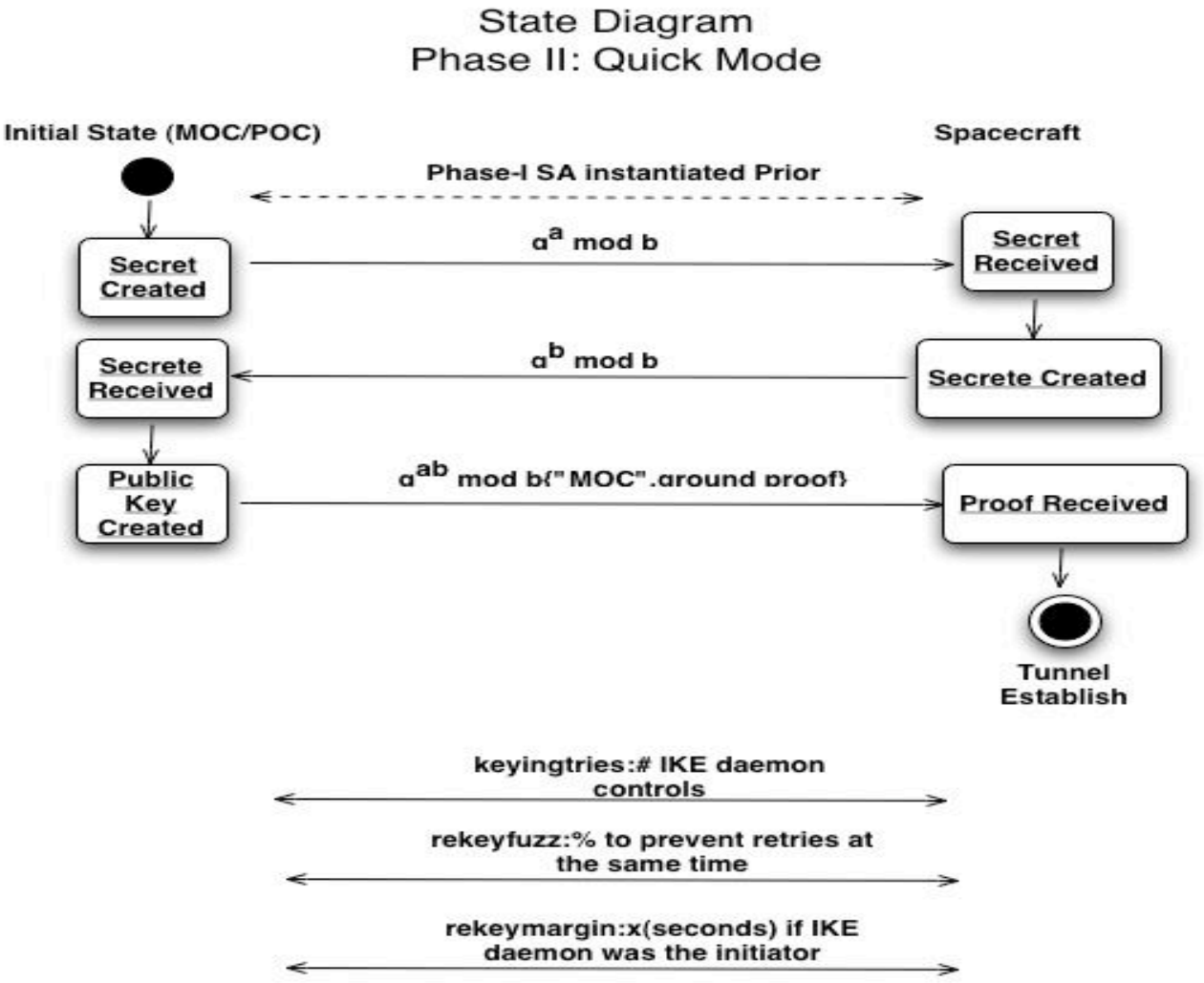


Baseline is now OpenSwan. Freeswan was discontinued 4/22/2003 to support IKEv2 and linux kernel 2.6

5.3 Quick Mode

One of the prior modes has to be established FIRST before you can use Quick Mode. Once an IKE SA is set up between Alice and Bob, either Alice or Bob can initiate and IPsec SA through the phase 2. The initiator of a phase 2 SA does not have to be the same party that initiated the phase 1 SA. This exchange establishes and ESP and/or AH SA, which involves negotiating crypto parameters, *optionally* doing a Diffie-Hellman exchange (if PFS is desired) and negotiating what traffic will be sent on the SA. Quick mode main points are:

- Establishes an ESP and/or AH SA, which involve negotiating crypto parameters, optionally doing a Diffie-Hellman exchange (if PFS is desired), and negotiating what traffic will be sent on the SA.
- Quick Mode has a traffic selector that can restrict traffic sent on that SA., by IP address, transport protocol and/or port number.



Note: There is an option not to re-key if you are not the initiator and there was recent traffic on the existing connection.

Baseline is now OpenSwan. Freeswan was discontinued 4/22/2003 to support IKEv2 and linux kernel 2.6

Cisco offers a mode known as “Dangling Mode” which allows IKE to disable itself and re-enable itself as needed. This is not an RFC but a “feature” of Cisco’s product line.

In conclusion Main Mode is more robust in terms of configurations and because of its intricate message scheme, it is more difficult to thwart which is why FSW and Ground Systems chose this configuration for IPSec communications.

6. Internet Key Exchange (IKE)

IKE (Internet Key Exchange) is defined as follows:

Security Associations are used with IPsec to define the processing done on a specific IP packet. An outbound packet produces a hit in an SPD (Security Policy Database), which is an entry to one or more Security Associations (SA)s. There is no SA associated with the Security Parameter Index (SPI). so it is necessary to create one. This is where IKE comes into play. The premise of IKE is to establish shared security parameters and authenticated keys, SAs, between IPsec peers.

6.1 Version I.

The IKE protocol is a hybrid of Oakley and SKEME protocols and operates inside a framework of ISAKMP. Respectively RFCs 2407,2408, and 2409 used to encompass IKE v1. Informational Messages is what RFC refers to as cookies, which contain state information that is passed during the aggressive and main modes. Throughout this explanation Oakley implementation refers to modes of operation, ISAKMP refers to phases of operation. The two are used interchangeably. Aggressive and Main Mode are both Phase I exchanges.

There are 8 variants of Phase I for IKE because there are 4 authentication methods (original public key encryption, revised public key encryption, public key signature and pre-shared keys). There is a Main and Aggressive Mode for each as well, hence there are many options. The messages are sent via UDP and require no response or confirmation upon their arrival. It should be noted they do contain state information in Phase II during the negotiation of a SA tunnel. It should also be noted by using the same key and not using auto key generation, PFS (Perfect Forward Secrecy) is not attainable.

6.2 Version II.

IKEv2 is the consortium of these protocols into one methodology with some added features. It strengthens the arguments of the ISAKMP architecture and does not focus as much on the implementation of the pieces that define the ISAKMP architecture. Some of the features that were new that encompass this proposal are NAT traversal, Legacy Authentication, and remote address acquisition.

NAT traversal implies being able to instantiate an IPSEC tunnel across a NAT, which is not “truly” feasible today. In early IETF’ s documents this was defined in a BOF entitled NGISec Next Generation IPsec, which allows an IPsec tunnel to traverse a NAT gateway. This is one of many features that IKEv2 will include in future revisions. I have not seen many vendors that support this yet in their devices .The closest thing that comes to mind is One-To-One Nat-ing that many vendors now support in their devices. This feature is implemented in phase I. In version II as well there something referred to as “Continuous Channel Mode” which is a mode that allows the command channel to always be up and enabled. At the same time when the command channel dies so does the security association as well without user intervention.

In version two you have a cipher suite called AES-XBC-MAC-96 that allows for legacy implementations of the first version of the protocol that was implemented. Another version that is being considered is the AES-XCBC-MAC-96. This suite addresses some of the known issues with fixed block sizes that CBC was intentionally designed for but since IP is variable length, this MAC helps address those issues. So if your messages are fixed lengths the present architecture is sufficient with CBC but with streamed cipher suites and variable length IP datagrams that are not fixed, it was proven to be possible to compromise this type of data because of the aforementioned CBC methodology. Cryptologist found that if your data exceeded the length of protection, it was possible for the algorithm to either fail or encrypt only portions of the data packet or packets in some cases. AES-XCBC-MAC-96 was proposed to resolve the threat of this vulnerability.

6.3 Version Summary.

IKEv1 vs IKEv2 (General)

Future, Beyond the security

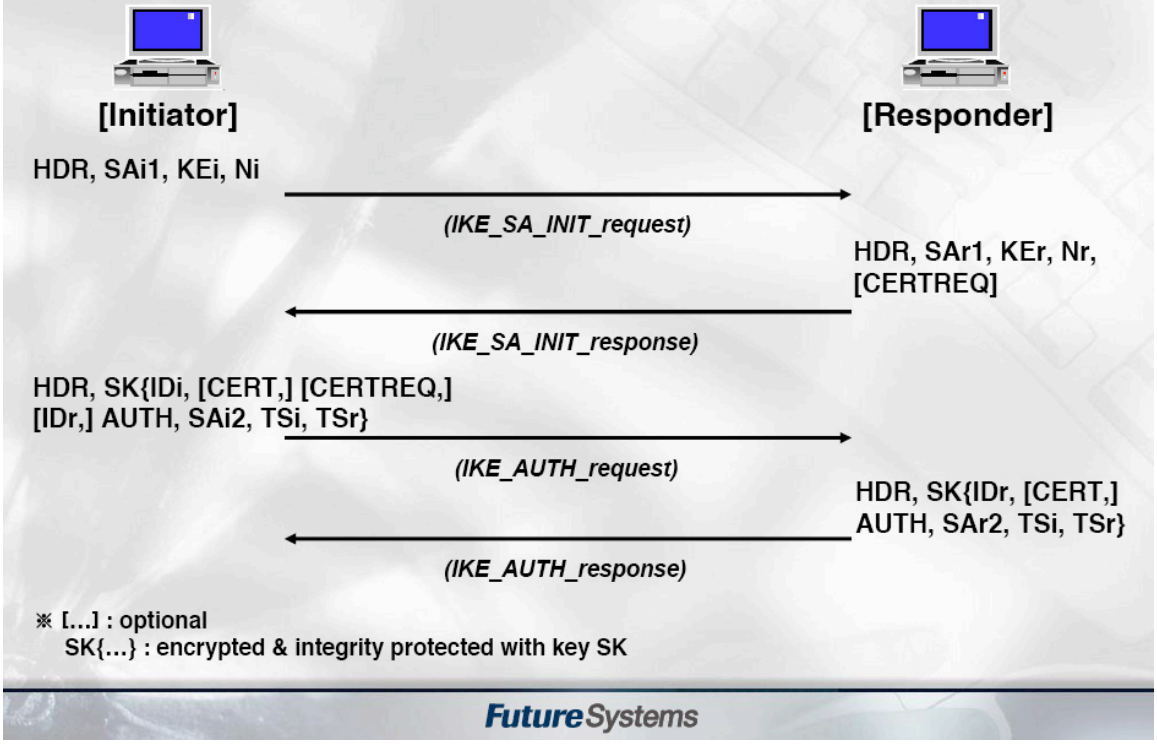
	IKEv1	IKEv2	Key word
RFC Document(s)	RFC 2407/2408/2409	RFC XXXX	*Merging*
Protocol Phase 1 Phase 2	2 Phase 6 or 3 messages 3 messages	2 Phase 4 messages(1) 2 messages	*Simplicity*
Authentication type	Signature, Pre-shared, (Revised) Public-key	Signature, Pre-shared	
SA negotiation	Responder's selection for Initiator's proposal(2)		
Identity Hiding	Optional(3)	Always	*Security*
Perfect Forward Secrecy	Yes(optional)	Yes(optional)	
Anti-DoS	No	Yes(optional)	
Input of HASH	A part of messages	All messages	
Reliability	Unreliable	Reliable(4)	*Reliability*
Backward compatibility	No	Yes	*Etc.*
Legacy Authentication	-	EAP(5)	
Remote address acquisition	-	CP payload	

(1) Allow to piggyback for CHILD_SA, (2) In IKEv2, redefine & simplify substructure of Proposal, (3) only in Main mode,
(4) All messages are acknowledged(request-response pair) and sequenced, (5) Extensible Authentication Protocol

FutureSystems

Phase 1 of IKEv2

Future, Beyond the security



※ [...] : optional
SK{...} : encrypted & integrity protected with key SK

FutureSystems

Phase 2 of IKEv2

[Initiator]

[Responder]

HDR, SK{[N], SA, Ni, [KEi], [TSi, TSr]}

(CREATE_CHILD_SA_request)

HDR, SK{SA, Nr, [KEr], [TSi, TSr]}

(CREATE_CHILD_SA_response)

HDR, SK{[N], [D], [CP],…}

(Informational_Exchange_request)

HDR, SK{[N], [D], [CP],…}

(Informational_Exchange _response)

※ [...] : optional
SK{...} : encrypted & integrity protected with key SK

IKEv1 vs IKEv2 (Payload format)

	IKEv1	IKEv2	Notes
Changed	✓ ISAKMP Header ✓ Generic Header ✓ Security Association ※ Transform ✓ Key Exchange ✓ Identification ✓ Hash ✓ Signature ✓ Notification ✓ Delete	✓ IKE Header ✓ Generic Header ✓ Security Association ※ Transform ✓ Key Exchange ✓ Identification ✓ Authentication ✓ Notify ✓ Delete	- Redefine Flags field - Add Critical bit field - Omit DOI/Situation ※ rename/restructure - Add DH Group # field - Omit field for DOI - Merged & Add Auth Method field - Omit DOI field - Omit DOI field
No changed	✓ Certificate ✓ Certificate Request ✓ Nonce ✓ Vendor ID	✓ Certificate ✓ Certificate Request ✓ Nonce ✓ Vendor ID	N/A
Added in IKEv2	N/A	✓ Traffic Selector ✓ Encrypted ✓ Configuration ✓ EAP payload	- For TS negotiation - For IKE msg. protection - Request for conf. info. - For EAP authentication

Features of updated ESP & AH

Future, Beyond the security

Added/Changed Features		ESP	AH
Security service	- Confidentiality-only service : now a May, not a Must	○	X
SPI	- Modified to specify a uniform algorithm for SAD lookup for unicast and multicast SAs.	○	○
Sequence num.	- Added a new option for a 64-bit sequence number (ESN) for very high-speed communications.	○	○
Mandatory algo.	- Moved references to mandatory algorithms to a separate document.	○	○
Combined mode algorithms	-Broadened model to accommodate combined mode algorithms → Payload data, ICV, Algorithms, Packet processing	○	X
TFC padding	- Added requirement to be able to add bytes after the end of the IP Payload, prior to the beginning of the Padding field.	○	X
Next Header	- Added requirement to be able to generate and discard dummy padding packets (Next Header = 59)	○	X

Future Systems

 [1]

[1] Future Systems, http://www.ipv6.or.kr/2003jeju/material/SIII-2.pdf, 2004

7. Security Algorithms

There are a plethora algorithms that are compliant with NPG2810.1. Which every algorithm is used it must be NIST compliant. AES and SHA are certified by NIST as being FIPS compliant. Based on the IP Security Handbook these algorithms were nominated as a consideration. This is not definitive by any means, but the algorithms discussed in this section are a viable candidate for consideration.

7.1 Encryption Algorithms (Confidentiality)

AES will be discussed because it is in compliancy with NIST standards and NASA Policy Guideline standards as well. The AES standard specifies the Rijndael algorithm, a symmetric block cipher that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits. Rijndael was designed to handle additional block sizes and key lengths, however they are not adopted in this standard. Throughout the remainder of this standard, the algorithm specified herein will be referred to as “the AES algorithm.” The algorithm may be used with the three different key lengths indicated above, and therefore these different “flavors” may be referred to as “AES-128”, “AES-192”, and“AES-256”.

The key length of the implementations is directly proportional to the hash key lengths, which is approximately one half the size of the hash key lengths. That is why respectively you have SHA-256, SHA-384 and SHA-512. If these hashes are going to be used in an algorithm for the purposes of authentication, a HMAC will probably be utilized instead of the hash by itself. An HMAC is a hashed key instead of a key that has been hashed to be used for a message authentication control scheme.

The reason for the origination of AES was to take over DES (Data Encryption Standard). DES symmetric key length is 56 bits, resulting a key space that contains only 2^56 possible different keys. In terms of brute force attacks, this attack is successful on small key space algorithms. DES is susceptible to brute force attacks. DES was first publicly cracked in 1997 at an RSA Challenge: five-month effort. All subsequent attempts are now taking less time thanks to Moore’s law and clustering of computers. By the summer of 1998 a brute force attack was successful on DES using a single processor.

This is why triple DES was created to expand the key length to approx 192bits in theory but subtract 24 bits for parity and you get actually 168 bits. Two to the 168 is a large key space but with Moore’s law and dual processor computers that are being sold as home PC Agencies like NIST new this was only a short term solution, hence the birth of AES.

AES Additional Information & Fact Sheet (NIST)"

Assuming that one could build a machine that could recover a DES key in a second (i.e., try 255 keys per second), then it would take that machine approximately 149 thousand-billion (149 trillion) years to crack a 128-bit AES key. To put that into perspective, the universe is believed to be less than 20 billion years old."

"Barring any attacks against AES that are faster than key exhaustion, then even with future advances in technology, AES has the potential to remain secure well beyond twenty years."[2] This may be mathematically true but if we ever reach quantum computing where one bit can represent a zero and/or one autonomous of the common substrates i.e. Si, Ga used for transistors today, all of these algorithms might just become useless.

In the interim each mission should decide on a couple factors regarding the selection of the keys and they are following: the mission lifecycle, and the consequences of having previous command and data compromised

[2] Arctisoft, http://www.articsoft.com/aes.htm, 2003

after a mission lifecycle has been expired. If the future missions rely on legacy commanding structure and source code, those exploits that were discovered in a mission 5-10 years ago may be the same types of exploits possible in present missions that are using those prior missions as baselines. The key selection process must take these factors into account.

7.2 Hash Algorithms (Host Authentication and Data Integrity)

Hash algorithms pre-dominant purpose is to provide authentication. Usually the authentication is that of a host (computer, workstation, server) and not of the user. Hash algorithms can also be used to validate the integrity of data as well. Hash algorithms have evolved because the key length increased and doubled in size. MD5 was sufficient with DES implementations of encryption because the key sizes were 54bits to 168 bits in length. Earlier implementation of encryption only used a 40-bit key. In Checkpoint Firewall, one the most trusted devices in the industry of network security, used a protocol called FWZ which was proprietary to Checkpoint but utilized a 40-bit key! The protocol was not open to scrutiny by the cryptanalysis community so it was more so "security by obscurity".

With succession of AES, the key lengths are now 128, 192, and 256 bits in length. The hash key lengths are typically double in size of the encryption key lengths. That is why in the successor to MD5, Secure Hash Algorithm –1 (SHA-1) now supports key length of 256, 383, and 512 bits long. The only place MD5 is utilized safely is with an HMAC (Hashed Message Authentication Code). The algorithm for HMAC is sufficient for IPsec and commercial file integrity applications such as Tripwire. An HMAC is a hashed key instead of a key that is hashed used for message authentication control scheme.

HMAC algorithm guarantees the following:

- Collision resistance making it infeasible to find two inputs that yields the same output.
- And attacker that does not know the Key K cannot compute the proper digest (K,x) for data x, even if the attacker can see the value of the digest (K,y) for arbitrary numbers of inputs y, with y not equal to x.

The order of operation for the HMAC algorithm is as follows:

1. It first pads the key with 0 bits to 512 bits. If the key is longer than 512 bits, then HMAC first digest the key, resulting in a 128 bits or 160 bits (depending on the size of the output of the digest function).
2. It then pads the result out to 512 bits.
3. Afterwards it XORs the padded key with a constant string of octets of value 36 base 16, concatenates it with the message to be protected and computers a message digest.
4. Lastly it XORs the padded key with a different constant string of octets of value 5c base 16 concatenates that result of the first digest, and computes a second digest on the result.

SHA-1 was chosen to be the hash algorithm GPM would use but the hash key length only needs to be the size necessary to comply with the AES key length of 128 bits.

In terms of the National Institute of Standards and Technology (NIST) SHA-224 has been certified. It was announced on February 28, 2004 the standard FIPS 180-2 Change Notice, which specifies the SHA-224 one-way hash function. One-way hash functions are also known as message digests. SHA-224 is based on SHA-256, the 256-bit one-way hash function already specified by NIST[SHA2]. Computation of a SHA-224 hash

value is two steps. First, the SHA-256 hash value is computed, except that a different initial value is used. Second, the resulting 256-bit hash value is truncated to 224 bits.

NIST is developing guidance on cryptographic key management, and NIST recently published a draft for comment [NISTGUIDE]. Five security levels are discussed in the guidance: 80, 112, 128, 192, and 256 bits of security. One-way hash functions are available for all of these levels except one. SHA-224 fills this void. SHA-224 is a one-way hash function that provides 112 bits of security, which is the generally accepted strength of Triple-DES [3DES].

Usage Considerations are the following:

Since SHA-224 is based on SHA-256, roughly the same amount of effort is consumed to compute a SHA-224 or a SHA-256 digest message digest value. Even though SHA-224 and SHA-256 have roughly equivalent computational complexity, SHA-224 is an appropriate choice for a one-way hash function that provides 112 bits of security. The use of a different initial value ensures that a truncated SHA-256 message digest value cannot be mistaken for a SHA-224 message digest value computed on the same data.

Some usage environments are sensitive to every octet that is transmitted. In these cases, the smaller (by 4 octets) message digest value provided by SHA-224 is important.

These observations lead to the following guidance:

- When selecting a suite of cryptographic algorithms that all offer 112 bits of security strength, SHA-224 is an appropriate choice for a one-way hash function.
- When terseness is not a selection criterion, the use of SHA-256 as a preferred alternative to SHA-224.[3]

Other hashes exist such as Lamport's Hash and many others, but until an authoritative body such as NIST and "business" protocol authoritative body such as the IETF inducts this algorithm, it is not popular or common in present implementation of group 1 and group 2 of the IPsec protocols.

In conclusion the hash algorithm that is selected by a mission first must take into the account the key exchange algorithm and key length. The key-length is proportional to the hash key length. Therefore, when a mission decides on an encryption algorithm they must also understand the impact on the hash algorithm will play in the overall computation.

8.0 Conclusion

In order for IPSec to be integrated into any communication system in FSW, knowledge of the protocol and configuration features must be taken into perspective with respect the impact on memory, cpu utilization, task priority, TCP/IP stack implementation, and kernel modularity. With respect to ground systems the use cases are key to operations concepts modeling for producing solutions for Ops concepts that can be deduced and configured in accordance to the spacecraft's ICD for space-to-ground link. The team works cohesively to devise a methodology of commanding telemetry of the spacecraft to satisfy onboard requirements for FSW and MOC/POC requirements for operation concepts.

[3] IETF, http://www.ietf.org/internet-drafts/draft-ietf-pkix-sha224-01.txt, 2004

IPSec was an option considered to be used onboard the spacecraft and to establish secure tunnels to the ground systems network premise equipment. The challenges lied ahead in the implementation of retrofitting the source for flight and integrating the debugging module KLIPS into your RTOS kernel. The other concern was compatibility with the ground segments so that we are not re-inventing IPsec but in fact following as COTS products have and the RFCs state defined by the IETF. These issues mentioned prior are not the focus of this document for they are implementation specific.

FSW was not able to draw a conclusion on all items of IPSec configuration but what was considered was the following: using Main Mode with pre-shared keys with a cipher suite of SHA-1 for authentication and AES-128 bit for encryption for the ESP protocol is a viable candidate. The keys will reside on the spacecraft and at the ground station. We will not be using a PKI (Public Key Infrastructure) of any sort. There is nothing precluding that a mission could not use 3DES and different HASH algorithms that are compliant. The reiteration is expressed again that the NPR 2810.1 policies and guidelines be adhered to in what ever selection a mission chooses to take and follow.

9.0 Appendix

9.1 Performance Characterization

The purpose of this section is to document the FSW aspect of the GPM trade to add a processor to the Comm. Card. The processor in the Comm. Card was investigated in order to support the IP Security requirement.

Several assumptions were made when determining the processor performance:

1) GPM has an IPSec requirement that includes authentication and decryption.
2) The AES-128 encryption algorithm was used in order to estimate the decryption aspect of the processor performance.
3) Only the uplink at a maximum speed of 64 kbps would have to authenticate and decrypted.
4) The Comm. card would have to simultaneously downlink a maximum of 4Mbps. The downlink data does not need to be encrypted except for possibly a CLCW. For the purposes of this exercise, a worst-case number of 1 MTU of encrypted downlink is used.
5) All of the HDLC framing is done in hardware.
6) MTU is assumed to be 1500 bytes (46-1500 bytes for least to maximum Ethernet payload.)
7) Core Services developed for the other GPM processors would be used on the Comm. Card.
8) The Comm. card would perform simple routing functions.
9) Other common functions such as table manager, file management, health and safety etc would be implemented on the Comm. Card.
10) 50% margin is required for memory and processing
11) Static key exchange is assumed

Memory Requirements

In order to estimate the amount of non-volatile and volatile memory required for the Comm. card, the application specific software needed to be estimated. An implementation of the AES-128 algorithm that was developed by Dr. Rijndael, was used for benchmarking. The software is called Crypto and was copyrighted by Dr. Brian Gladman. The software required approximately 50Kbytes of non-volatile memory (code and data). No volatile memory requirements are available, therefore, an estimate of 100Kbytes is assumed. In addition to the decryption/encryption software, routing software, core services and other common functions were added in order to provide a total comm. card processor memory estimate. The details of the estimate can be found under "GPM Flight Software Documents".:[4]

In order to provide a sanity check to our estimate and provide an upper bound to the memory requirement, a COTS product made by Snapgear was evaluated. Alan Cudmore was able to get a modified version of the Snapgear code to execute on a MCF5307 Coldfire evaluation board under the Linux. The software required a little less than 1 Mbyte non-volatile memory and 1 Mbyte of volatile memory.

The following summarizes the FSW Comm. Processor memory requirements:

Non-volatile (with 50% margin): 2Mbyte
Volatile (with 50% margin): 2Mbyte

Processor Performance Requirement

[4] GPM Flight Software Documents, http://fsw.gsfc.nasa.gov/internal/gpm/, 2004

The two driving requirements from a processor performance point of view are assumed to be the decryption/encryption algorithm and the data movement aspect of the Comm. Card.

For the purposes of the performance estimate, the following assumptions are made:

1) The decryption algorithm must be able to decrypt 64 kbps.
2) ColdFire can be run at least 36/18 for 3x performance increase.
3) NIC and HDLC memory mapped at same speed as SRAM 32bits wide.
4) 3 BCLK0 cycles to read or write external SRAM/NIC/HDLC on the SDN board.
5) No PCI bus interface.
6) Instructions are in cache.

In order to get a handle on the decryption processing requirements, benchmarking of an implementation of the ACE-128 algorithm was performed. The benchmarking was performed using the Crypto algorithm on a prototype Subsystem Data Node (SDN) that uses a Motorola Coldfire RHCF-5208 processor. The operating system used was RTEMS. The prototype board's clock speed is 12Mhz with memory access of 6 Mhz. The benchmarking was performed with various size packets from the smallest (64 bytes) to the largest (1500 bytes). It turns out that the performance of the decryption algorithm was much worst with the smaller 64-byte packets than with the larger packets. Results of the benchmark are shown in table 1.1 show the results. Worst case, it would take 631ms to decrypt the uplink which violates the 50% margin requirement.

bytes	#packets per 64kbps uplink	time (encrypt & decrypt)	Time to decrypt only	Time to decrypt each packet (ms)	total time to decrypt 64kpbs worth of packets (ms)
1500	5.3	6.928	3.464	40.753	**217**
1000	8	4.798	2.399	28.224	**226**
500	16	2.725	1.3625	16.029	**256**
100	80	1.045	0.5225	6.147	**492**
64	125	0.858	0.429	5.047	**631**

Table 1.1

If the clock and memory performance of the Coldfire was increased to 36/18, an increase of 3 times the performance of the SDN prototype, it would take approximately **210 ms** to decrypt the uplink, worst case (630ms / 3).

In addition, data must be moved from the NIC to the HDLC encoder FIFO at a maximum rate of 4Mbps or 125,000 32bit words. The following code instructions and their performance numbers are as follows (note that BCLK0 = 55ns):

```
Code    a0 = NIC address pointer, a1 = FIFO address
d1   count

..LMEMCPY:
        move.l (a0)+,(a1)    ; 112ns + 165ns (read) + 165ns (write)
        sub.l #1,d1          ; 28ns
        bcc .LMEMCPY         ; 28ns (branch taken)
```

The result is that it takes 498ns per 32bit memory-to-memory move or ~63ms to move 125,000 32bit words.

Hence, the Comm. card processor operating at 36/18 would be able to decrypt/encrypt and move the 4Mbps downlink in 273ms. Since the FSW is required to have a 50% margin, the comm. card is left with 227 ms for other processing that is sufficient.

In conclusion, the FSW team believes that a Coldfire microprocessor operating at 36/18 Mhz speed with 2 Mbytes of volatile and 2 Mbytes non-volatile will meet the requirements of the GPM communications subsystem.

9.2 References

1Doraswamy, Naganand, Harkins, Dan. IPSEC The Security Standard for the Internet, Intranets, and Virtual Private Networks. New Jersey:Prentice Hall, 1999.

2 Kaufman, Charlie, Perlman, Radia, Spencer. Mike. Network Security: Private Communication in a Public World 2nd Edition. New Jersey:Prentic Hall, 2002.

3 Code 297. NPG2810.1, 2002.
Note: NPR 2810.x was not published at the fabrication of this document

4 NASA GSFC-Code 588, Computer Science Corporation, IP-In-Space Security Handbook, NASA GSFC:Code 297, 2001.

Internet Engineering Task Force (IETF)
RFCs:

- ISAKMP RFC 2408
- IKEv.1 RFC 2409
- Internet DOI RFC 2407
- HMAC-SHA RFC 2403
- HMAC-SHA-96 RFC 2404
- IPSec-AH RFC 2402
- IPSec-ESP RFC 2406
-

IKEv.2
Note:IKEv.2 is still in draft mode and does not have an associated RFC number.[5]

9.3 Vendor Specific Implmentations

Timers are vendor specific in terms of their implementation. The only timers that are the same across manufactures are ISKMP lifetime /IKE lifetime and IPsec lifetime timers. The timers we are interested in are at a lower level of granularity. All vendors implement these timers differently but they are alike enough for interoperability and they are also inline with the RFC boundaries defined in 2409, and 3706.

9.3.1 OpenSwan/FreeSwan's Timing Implementation

[5] IETF, http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ikev2-14.txt, 2004

For automatic keying the time parameters are the following in a typical opensource implementation (Pluto is the ISAKMP/IKE daemon in this example):

--ikelifetime *seconds*
how long Pluto will propose that an ISAKMP SA be allowed to live. The default is 3600 (one hour) and the maximum is 28800 (8 hours). This option will not affect what is accepted. Pluto will reject proposals that exceed the maximum.

--ipseclifetime *seconds*
how long Pluto will propose that an IPsec SA be allowed to live. The default is 28800 (eight hours) and the maximum is 86400 (one day). This option will not affect what is accepted. Pluto will reject proposals that exceed the maximum.

--rekeymargin *seconds*
how long before an SA's expiration should Pluto try to negotiate a replacement SA. This will only happen if Pluto was the initiator. The default is 540 (nine minutes).

--rekeyfuzz *percentage*
maximum size of random component to add to rekeymargin, expressed as a percentage of rekeymargin. Pluto will select a delay uniformly distributed within this range. By default, the percentage will be 100. If greater determinism is desired, specify 0. It may be appropriate for the percentage to be much larger than 100.

--keyingtries *count*
how many times Pluto should try to negotiate an SA, either for the first time or for re-keying. A value of 0 is interpreted as a very large number: never give up. The default is three.

--dontrekey
A misnomer. Only re-key a connection if we were the Initiator and there was recent traffic on the existing connection. This applies to Phase 1 and Phase 2. This is currently the only automatic way for a connection to terminate. It may be useful with Road Warrior or Opportunistic connections.
Since SA lifetime negotiation is take-it-or-leave it, a Responder normally uses the shorter of the negotiated or the configured lifetime. This only works because if the lifetime is shorter than negotiated, the Responder will re-key in time so that everything works. This interacts badly with --dontrekey . In this case, the Responder will end up re-keying to rectify a shortfall in an IPsec SA lifetime; for an ISAKMP SA, the Responder will accept the negotiated lifetime.

If you use manual keying then the SA NEVER expires unless you physically remove it. Manual keying is also of course less secure for many reasons that will be discussed in this document.

9.3.2 Cisco's Timing Implementation

- In Cisco's implementation ISKMP/IKE Lifetime refers to only Phase I i.e. Main Mode of the IKEv.1. IPSec Lifetime refers to only Phase II i.e. Quick Mode.
- There are a total of 6 messages that have to be sent in Main Mode to establish a tunnel. Cisco's implementation is that each message has 5 times to be re-transmitted before the security association fails completely.
- Each message has 10 seconds to be acknowledged in order for it to proceed to the next state (see state diagram in section 2.0). Therefore worst case would be taking the entire 10 seconds per state each time and having to re-transmit 5 times for each state in terms of total latency.

For FSW this is unacceptable and we would not allow this to happen. In software we would default back to another SA while this new SA tries to establish. When the new SA is established the old one will be deleted automatically because of the IKE protocol by default configuration. Bare in mind that the SA that has to be the backup cannot be on the same MAC interface. It has to be affiliated with another MAC PHY physical interface. Keep in mind you can have multiple SAs in the SADB (Security Association Database) instantiated simultaneously; the SAs may not be defined with the same name though. Also we have the ability to terminate our SAs via commands manually as well.

It was confirmed by Cisco that FreeSwan/OpenSwan is compatible with Cisco's product line as well. The only question is the compatibility of IKEv.2 which they believe will have bugs during the first few instantiations but afterwards those issues will be resolved and there should not be a problem with regards to interoperability.

Cisco's has also stated they will support IKEv.1 and version 2 in their product line as well. They will continue to support both of the keying methodologies in all of their premise equipment.

9.4 Glossary

- **Authentication Header (AH)**: The IPSec header used to verify that the contents of a packet haven't been modified in transit.*
- **Authentication**: The process of validating the claimed identity of an end user or a device such as a host, server, switch, router, and so on.*
- **Bandwidth**: 1) A range within a band of frequencies or wavelengths. (2) The amount of data that can be transmitted in a fixed amount of time. For digital devices, the bandwidth is usually expressed in bits per second (bps) or bytes per second. For analog devices, the bandwidth is expressed in cycles per second, or Hertz (Hz).[6]
- **Cisco:** A vendor of routers, hubs, switches, firewalls, and related products.
- **Data Encryption Standard (DES)**: A secret key cryptographic scheme standardized by National Institute of Standard and Technology (NIST).*
- **Data Link Layer:** The services in the OSI protocol stack (layer 2 of 7) that manage node-to-node transmission.
- **Decryption**: A method of unscrambling encrypted information to make it legible.*
- **Denial of Service (DoS) Attack**: Any action that prevents any part of a network or host system from functioning in accordance with its intended purpose.*
- **Digital Certificates**: A message, digitally signed with the private key of a trusted third party, stating that a specific public key belongs to someone or something with a specified name and set of attributes.*
- **Digital Signature**: A string of bits appended to a message (an encrypted hash) that provides authentication and data integrity; typically this term applies only to signatures generated using public key encryption.*
- **Encapsulating Security Payload (ESP)**: The IPSec protocol that provides the security services of confidentiality, traffic-flow confidentiality, connectionless integrity, data origin authentication, and an anti-replay service.*
- **Encryption**: A method of scrambling information in such a way that is not readable by anyone except the intended recipient, who must decrypt it to read it.*
- **Firewall**: A system, based on either hardware or software, that applies rules to control the type of networking traffic between two networks.*
- **Hash function**: A mathematical computation that results in a fixed-length string of bits (digital code) from an arbitrary size input; a one-way hash function is not reversible to produce the original input.*
- **Hash**: The resulting bits from a hash function.*

[6] Webopedia, http://inews.webopedia.com/TERM/B/bandwidth.htm, 2003

- **Internet Engineering Task Force (IETF)**: A standards body whose focus is to design protocols for use on the Internet. Its publications are called Requests for Comments (RFCs).*
- **Internet Key Exchange (IKE)**: The protocol that specifically defines the negotiation and keying exchange for IPSec.*
- **Intrusion Detection System (IDS):** A system that tries to identify attempts to hack or break into a computer system or to misuse it. IDS's may monitor packets passing over the network, monitor system files, monitor log files, or set up deception systems that attempt to trap hackers.
- **Internet Protocol (IP) Telephony**: IP telephony enables people to use the data network as the transmission medium for telephone calls. For users who have free or fixed-price Internet access, Internet telephony software essentially provides free telephone calls anywhere in the world. Internet telephony products are sometimes called IP telephony, Voice over the Internet (VOI) or Voice over IP (VoIP) products. When the transport layer is the public Internet or the Internet backbone from a major carrier, it is generally called "IP telephony" or "Internet telephony." However, the terms IP telephony, and VoIP are used interchangeably.[7]
- **Internet Protocol (IP) Security Protocol (IPSec)**: A set of network layer protocols that collectively can be used to secure IP traffic.*
- **Internet Protocol (IP)**: IP is connectionless and uses higher layers protocols such as UDP and TCP to enable "sessions" across the network, including voice calls.[8] IP has robust signaling, addressing, and routing functionality, integrates well with current data applications and is the most ubiquitous networking protocol. IP also has the distinct advantage of being a Layer 3 protocol, so it can leverage the benefits of a layer 2 Frame Relay or ATM networks. IP runs all the way to the desktop for the greatest flexibility in supporting new Web-based IP applications, open IP-based PBXs, and IP telephones.
- **Jitter**: Jitter is a variation in the time of arrival of received signals. Increased jitter makes it harder to tell when a packet is missing or just late.
- **Latency**: The time from when words are spoken until they are heard at the other end. Latency greater than 150 milliseconds is unacceptable in most cases.[9]
- **Message digest**: The value returned by a hash function (same as hash).*
- **MultiProtocol Label Switching (MPLS)**: A short fixed-length label is generated that acts as a shorthand representation of an IP packet's header. Subsequent routing decisions (made by Label Switched routers) are made based on the MPLS label and not the original IP address. This new technology allows core network routers to operate at higher speeds without needing to examine each packet in detail, and allows more complex services to be developed, allowing discrimination on a QoS basis.[10]
- **National Institute of Standards and Technology (NIST)**: An agency of the U.S. government that establishes national technical standards.*
- **Network Address Translation (NAT)**: The process of converting one IP address to another IP address; often used to connect networks with a private address space to the Internet.*
- **Network Intrusion Detection System (NIDS):** IDSs that operate on network data flows.
- **Network Layer:** The services in the OSI protocol stack (layer 3 of 7) that provide internetworking for the communications session.
- **Open System Interconnection (OSI)**: An ISO standard for worldwide communications that defines a networking framework for implementing protocols in seven layers. Control is passed from one layer to the next, starting at the application layer in one station, and proceeding to the bottom layer, over the channel to the next station and back up the hierarchy.[11]
- **Physical Layer:** The services in the OSI protocol stack (layer 1 of 7) that provide the transmission of bits over the network medium.

---

[7] Dr. Harold J. Podell, *Introduction to Multiservice Networks: Security Architecture Issues*, Draft Version 9.0, Spring 2004.
[8] Dr. Harold J. Podell, *Selected Updates and Suggestions Pertaining to Enterprise and Network Security*, Draft Version 6, Fall 2003.
[9] Rick Kuhn, *Voice Over Internet Protocol (VOIP) Security*, Computer Security Division, NIST.
[10] http://www.interoute.com/glossary.html. 2003
[11] http://inews.webopedia.com/TERM/O/OSI.html. 2003.

- **Port numbers:** Ports are numbers ranging from 0 to 65,000, which allow transmissions to be sent directly to a particular piece of software which is 'listening' to the specified port on a particular machine. Port numbers under 1024 are "privileged" which are assignable to particular services only by the administrator of a machine.
- **Presentation Layer:** The services in the OSI protocol stack (layer 6 of 7) that provides conversion of codes and formats for the communications session.
- **Protocol**: A formal set of conventions governing the format and control of inputs and outputs between two communicating devices. This includes the rules by which these two devices communicate as well as handshaking and line discipline.
- **Public Key Infrastructure (PKI)**: A trusted and effective key and certificate management system.*
- **Quality of Service (QoS)**: It refers to the speed and clarity expected of a VoIP conversation.[12]
- **Real Time Control Protocol (RTCP):** RTCP is a companion protocol that is used to maintain QoS. RTP nodes analyze network conditions and periodically send each other RTCP packets that report on network congestion.[13]
- **Real-time Transport Protocol (RTP):** The Internet-standard protocol for the transport of real-time data, including audio and video. RTP is used in virtually all voice-over-IP architectures, for videoconferencing, media-on-demand, and other applications. RTP provides services such as payload type identification, sequence numbering, time stamping, and delivery monitoring to real-time applications.
- **Rivest Cipher 4 (RC-4)**: A variable-key-size stream cipher designed by Ron Rivest for RSA Data Security, Inc.*
- **Secure Hash Algorithm 1 (SHA1)**: A one-way hash algorithm designed by NIST that has a 160-bit digest.*
- **Security policy**: The set of rules and practices that regulate how an organization manages, protects, and distributes sensitive information.*
- **Session Layer:** The services in the OSI protocol stack (layer 5 of 7) that initiates and manages the communications session.
- **Simple Network Management Protocol (SNMP)**: A protocol used in network management for monitoring and configuring network devices.
- **Spoofing**: An attempt to gain access to a networked device by posing as an authorized user, device, or program.*
- **Stateful Firewall**: It opens packets between layer 2 and layer 3 of the OSI model to fully inspect it.
- **Time Division Multiplexing (TDM)**: An older, but still used, wide-area networking technology in which data is split, or multiplexed, into time-specific segments for transmission over a single path. The segments are then put back together, or de-multiplexed, at the other end of the path.[14]
- **Throughput**: The amount of data transferred from one place to another or processed in a specified amount of time. Data transfer rates for disk drives and networks are measured in terms of throughput. Typically, throughputs are measured in kbps, Mbps and Gbps.[15]
- **Transmission Control Protocol (TCP):** A communications protocol that ensures data is sent between computers on the Internet. It is a connection-oriented protocol and operates at Layer 4, Transport Layer of the OSI model.
- **Transport Layer:** The services in the OSI protocol stack (layer 4 of 7) that provides end-to-end management of the communications session.
- **Triple DES (3DES)**: An algorithm that uses DES and one, two, or three keys to encrypt/decrypt/encrypt the data.*
- **Tunnel**: A vehicle for encapsulating packets inside a protocol that is understood at the entry and exit points of a given network; also, a secure virtual connection through the Internet or an intranet.*
- **User Datagram Protocol (UDP):** A connectionless, unreliable, transport protocol, which provides multiplexing, and error detection for applications that require a low-cost protocol.

---

[12] Rick Kuhn, *Voice Over Internet Protocol (VOIP) Security*, Computer Security Division, NIST.
[13] http://www.techweb.com/encyclopedia/defineterm?term=RTP .2003.
[14] http://www.nwfusion.com/details/709.html .2003.
[15] http://inews.webopedia.com/TERM/T/throughput.html .2003.

- **Virtual Private Network (VPN)**: A network that is constructed by using public wires to connect nodes. For example, there are a number of systems that enable you to create networks using the Internet as the medium for transporting data. These systems use encryption and other security mechanisms to ensure that only authorized users can access the network and that the data cannot be intercepted.[16]
- **Wide Area Network (WAN)**: A network spanning a large geographical area. Its nodes can span city, state, or national boundaries. They typically use circuit provided by common carriers.[17]
- **Zero-day Attack:** Security vulnerability exploited in masses before it is reported.

9.5 Acronyms

- 3DES: Triple Data Encryption Standard
- ACL: Access Control List
- AES: Advanced Encryption Standard
- AH: Authentication Header
- DES: Data Encryption Standard
- DoS: Denial of Service
- ESP: Encapsulating Security Payload
- GRE: Generic Routing Encapsulation
- HIDS: Host Intrusion Detection system
- IDS: Intrusion Detection System
- IETF: Internet Engineering Task Force
- ISG: Information Security Governance
- I/O: Input/Output
- IOS: Internetwork Operating System
- IP: Internet Protocol
- IPSec: Internet Protocol Security
- IPv6: Internet Protocol version 6
- ITU-T: International Telecommunication Union
- Kbps: Kilo Bits Per Second
- LAN: Local Area Network
- MAC: Media Access Control
- MPLS: MultiProtocol Label Switching
- NAC: Network Admission Control
- NAT: Network Address Translation
- NIDS: Network Intrusion Detection System
- OOB: Out-Of-Band
- OS: Operating System
- OSI: Open System Interconnection
- PC: Personal Computer
- PKI: Public Key Infrastructure
- QoS: Quality Of Service
- RAS: Registration Admission and Status
- RC4: Rivest Cipher 4
- RSH: Remote SHell
- RTC: Real-Time Clock
- RTCP: Real-time Transport Control Protocol
- RTP: Real-time Transport Protocol

[16] http://inews.webopedia.com/TERM/V/VPN.html .2003.
[17] Jerry Fitzgerald & Alan Dennis, *Business Data Communications and Networking*, 1996

- SCCP: Signaling Connection Control Part
- SCP: Service Control Point
- SDP: Session Description Protocol
- SHA: Secure Hash Algorithm
- SNM: Secure Network Management
- SNMP: Secure Network Management Protocol
- SRTP: Secure Real-time Transport Protocol
- SSH: Secure SHell
- SSL: Secure Sockets Layer
- TACACS: Terminal Access Controller Access Control System
- TCP: Transmission Control Protocol
- TDM: Time-Division Multiplexing
- TLS: Transport Layer Security
- UDP: User Datagram Protocol
- USB: Universal Serial Bus
- VLAN: Virtual Local Area Network
- VoIP: Voice Over Internet Protocol
- VoWLAN: Voice Over Wireless Local Area Network
- VPN: Virtual Private Network
- WAN: Wide Area Network